Ch 1.2: Roundoff Errors & Computer Arithmetic

Computer arithmetic is different than our arithmetic, due to limited precision.

Example: Consider the following computations using a computer with 4 digit precision.

Arithmetic	Computer Arithmetic	Same?
$2 + 2 = 4$	$2.0 + 2.0 = 4.0$	Yes
$(4)(4) = 16$	$(4.0)(4.0) = 16.0$	Yes
$\left(\sqrt{3}\right)^2 = 3$	$\left(\sqrt{3}\right)^2 \cong (1.7321)^2 \cong 3.0002$	No

Finite Computer World

In the computational world, each number has only a finite number of digits.

Thus a computer can only accurately represent numbers that have finite decimal representations of a fixed length.

Hence only a subset of the rational numbers can be represented exactly by a computer.

The finite precision can result in inaccuracy of results as well as a variety of other problems, as we will see this semester.

Roundoff error is produced when a machine performs real number calculations, resulting in calculations with approximate representations of actual numbers.

Brief Review of Binary Number System

Base 10		Base 2
0	$0\cdot 2^1+0\cdot 2^0$	0
1	$0\cdot 2^1+1\cdot 2^0$	1
2	$1\cdot 2^1+0\cdot 2^0$	10
3	$1\cdot 2^1+1\cdot 2^0$	11
4	$1\cdot 2^2+0\cdot 2^1+0\cdot 2^0$	100
5	$1\cdot 2^2+0\cdot 2^1+1\cdot 2^0$	101
6	$1\cdot 2^2+1\cdot 2^1+0\cdot 2^0$	110
7	$1\cdot 2^2+1\cdot 2^1+1\cdot 2^0$	111
8	$1\cdot 2^3+0\cdot 2^2+0\cdot 2^1+0\cdot 2^0$	1000
9	$1\cdot 2^3+0\cdot 2^2+0\cdot 2^1+1\cdot 2^0$	1001
10	$1\cdot 2^3+0\cdot 2^2+1\cdot 2^1+0\cdot 2^0$	1010
11	$1\cdot 2^3+0\cdot 2^2+1\cdot 2^1+1\cdot 2^0$	1011
12	$1\cdot 2^3+1\cdot 2^2+0\cdot 2^1+0\cdot 2^0$	1100
13	$1\cdot 2^3+1\cdot 2^2+0\cdot 2^1+1\cdot 2^0$	1101
14	$1\cdot 2^3+1\cdot 2^2+1\cdot 2^1+0\cdot 2^0$	1110
15	$1\cdot 2^3+1\cdot 2^2+1\cdot 2^1+1\cdot 2^0$	1111

Similarly with fractional expansions. For example:

$$\frac{1}{2}=1\cdot\left(\frac{1}{2}\right)^1=0.1$$

$$\frac{1}{4}=0\cdot\left(\frac{1}{2}\right)^1+1\cdot\left(\frac{1}{2}\right)^2=0.01$$

Long Real Representations

The subset of rational numbers used by a computer contains numbers typically stored in a particular format, consisting of the fractional part together with an exponential part.

Many computers use a 64 bit (binary digit) representation for a real number, called a **long real**.

The first bit is a sign indicator (±), denoted *s*.

This is followed by an 11-bit exponent *c*, called the **characteristic**, and a 52-bit binary fraction *f*, called the **mantissa**. The base for the exponent is 2.

Note that $2^{52} \cong 4.5036 \times 10^{15}$ has 16 digits and 15 decimal digits (in the form displayed).

Thus 52 binary digits corresponds to between 16 and 17 decimal digits (base 10), and hence we have 16 decimal digit precision.

The 11-bit exponent gives range of 0 to $2^{11} - 1 = 2047$.

However, only using positive integers for the exponent excludes fractions with magnitude less than 1. ← $-1 < x < 1$

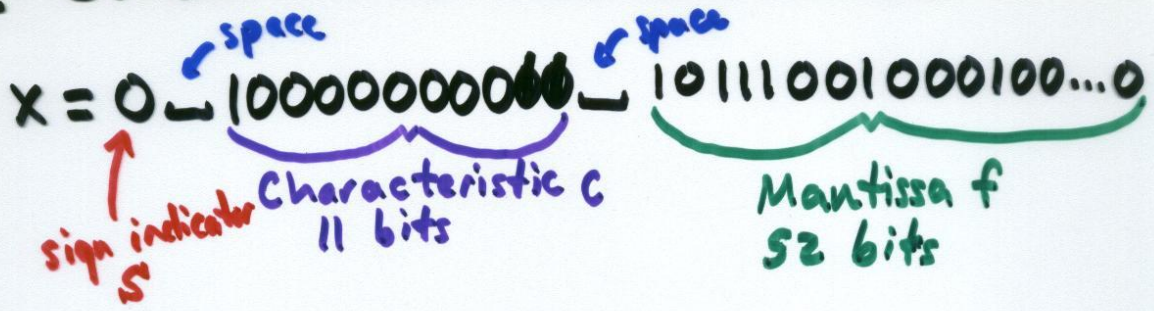
For these numbers, we need negative exponents.

Thus we subtract 1023 from the characteristic so that the range of the exponent is -1023 to 1024.

Using this system, a **floating point number** has the form

$$(-1)^s 2^{c-1023}(1+f)$$

Ex Consider the machine number



Thus $c = 10000000011$

$$= 1\cdot 2^{10} + 0\cdot 2^9 + 0\cdot 2^8 + \cdots + 0\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0$$

$$= 1024 + 2 + 1 \quad \text{(base 10)}$$

$$= 1027 \quad \text{(base 10)}$$

Hence the exponential part of x is

$$2^{c-1023} = 2^{1027-1023} = 2^4 = 16$$

The 52-bit mantissa is

$$f = 1\left(\tfrac{1}{2}\right)^1 + 0\cdot\left(\tfrac{1}{2}\right)^2 + 1\cdot\left(\tfrac{1}{2}\right)^3 + 1\cdot\left(\tfrac{1}{2}\right)^4 + 1\cdot\left(\tfrac{1}{2}\right)^5 + 0\cdot\left(\tfrac{1}{2}\right)^6 + 0\cdot\left(\tfrac{1}{2}\right)^7 + 1\cdot\left(\tfrac{1}{2}\right)^8 + 1\cdot\left(\tfrac{1}{2}\right)^{12}$$

$$= \frac{1}{2} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{256} + \frac{1}{4096}$$

Recall that the long real, or floating point form representation, has form

$$(-1)^s \cdot 2^{c-1023} \cdot (1+f)$$

Using the results on previous slide, we obtain

$$x = (-1)^0 \, 2^4 \left[1 + \tfrac{1}{2} + \tfrac{1}{8} + \tfrac{1}{16} + \tfrac{1}{32} + \tfrac{1}{256} + \tfrac{1}{4096}\right]$$

$$= 27.56640625$$

Note that the floating point format

$$(-1)^s \cdot 2^{c-1023} \cdot (1+f)$$

is similar to scientific notation format, with $\pm a.bcde \times 10^k$, where a, b, c, d are digits between 1 and 9 inclusive.

Recall the machine number x of the above example:

$x = 0\_10000000011\_1011100100010 0 \cdots 0$

Note that the next smallest machine number is

$y = 0\_10000000011\_1011100100001111 \cdots 1$

and the next largest machine number is

$z = 0\_10000000011\_1011100100010 0 \cdots 01$

It can be shown (as we did for x) that

$y = 27.56640624999999822364316059749535322$
$\quad 1893310546375$

and

$z = 27.56640625000000177635683940025049467781066894$
$\quad 53125$

See text. For comparison, recall that

$x = 27.56640625$

Note: x is the only representable number in (y, z) for computer.

The smallest positive machine number is

$$0\,\underbrace{00\ldots0}_{c}\,\underbrace{00\ldots01}_{f} = 2^{-1023}(1+2^{-52}) \cong 10^{-308}$$

(s)

The largest positive machine number is

$$0\,\underbrace{11\ldots1}_{c}\,\underbrace{11\ldots11}_{f} = 2^{1024}\underbrace{(2-2^{-52})}_{*} \cong 10^{308}$$

(s)

Geometric Series formula

$$\left[1 + 1\left(\tfrac{1}{2}\right) + 1\left(\tfrac{1}{2}\right)^2 + \cdots + 1\left(\tfrac{1}{2}\right)^{52} = \sum_{k=0}^{52}\left(\tfrac{1}{2}\right)^k = \frac{1-\left(\frac{1}{2}\right)^{53}}{1-\frac{1}{2}}\right.$$

$$\left. = \frac{1-2^{-53}}{\frac{1}{2}} = 2-2^{-52}\right] \quad *$$

Numbers occuring in calculations that have a magnitude less than $2^{-1023}(1+2^{-52})$ result in underflow and are generally set to zero.

Numbers greater than $2^{1024}(2-2^{-52})$ result in overflow and typically cause computations to halt.

To help better understand computational difficulties resulting from the use of a finite collection of numbers to represent all of the real numbers, we will use base 10, or decimal, representations of real numbers:

$$\pm 0.d_1 d_2 \cdots d_k \times 10^n,$$

where

$$1 \leq d_1 \leq 9 \quad \text{and} \quad 0 \leq d_i \leq 9, \; i = 2, 3, \ldots, k$$

Numbers of this form are called k-digit decimal machine numbers.

Suppose that y is a positive real number with the form

$$y = 0.d_1 d_2 \ldots d_k d_{k+1} d_{k+2} \ldots \times 10^n$$

and assume $10^{-308} < y < 10^{308}$.

The floating-point form of y, denoted by $fl(y)$, is obtained by terminating the mantissa of y at k decimal digits.

There are two ways of terminating the mantissa:

(1) Chopping: Simply chop off (or truncate) the digits $d_{k+1} d_{k+2} \ldots$ to obtain

$$fl(y) = 0.d_1 d_2 \ldots d_k \times 10^n$$

(2) Rounding: Add $5 \times 10^{n-(k+1)}$ to y and then truncate the result to obtain

$$fl(y) = 0.\delta_1 \delta_2 \ldots \delta_k \times 10^n$$

Note: $5 \times 10^{n-(k+1)} = 5 \times 10^{-(k+1)} \times 10^{n}$

$= (0.\underbrace{00\ldots0}_{k \text{ zeros}}5) \times 10^{n}$ (5 in the $(k+1)$-th position)

Thus when $d_{k+1} \geq 5$, we add 1 to d_k to obtain $fl(y)$:

$$y + [5 \times 10^{n-(k+1)}]$$

$$= 0.d_1 d_2 \cdots d_k d_{k+1} \cdots \times 10^{n} + 0.00\cdots0\,5 \times 10^{n}$$

$$\cong 0.d_1 d_2 \cdots (d_k + 1) \times 10^{n}$$

$$= 0.\delta_1 \delta_2 \cdots \delta_k \times 10^{n}$$

$$= fl(y)$$

This is commonly known as "rounding up" at k-th decimal place.

Also, when $d_{k+1} < 5$, we chop after d_k (i.e., round down):

$$y + [5 \times 10^{n-(k+1)}]$$

$$= 0.d_1 d_2 \cdots d_k \rfloor d_{k+1} \cdots \times 10^n + 0.00 \cdots 0 \rfloor 5 \times 10^n$$

$$\cong 0.d_1 d_2 \cdots d_k \rfloor (d_{k+1} + 5) \cdots \times 10^n$$

$$\cong 0.d_1 d_2 \cdots d_k \rfloor \times 10^n$$

$$= fl(y)$$

The error that results when replacing a number y with its floating point form $fl(y)$ is called **roundoff error**, regardless of whether rounding or chopping is used.

Ex Consider pi:

$\pi = 3.14159265\ldots = 0.314159265\ldots \times 10^1$

Using 5 digit chopping:

$fl(\pi) = 0.31415 \times 10^1 = 3.1415$

Using 5 digit rounding:

$fl(\pi) = 0.31416 \times 10^1 = 3.1416$

Note: Here $n=1$ and $k=5$, with

$$5 \times 10^{n-(k+1)} = [5 \times 10^{-(k+1)}] \times 10^n$$
$$= 5 \times 10^{-6} \times 10^1$$
$$= 0.000005 \times 10^1$$

Thus

$$\pi + 0.000005 \times 10^1$$
$$= [0.314159265\ldots \times 10^1] + [0.000005 \times 10^1]$$
$$= 0.314164265\ldots \times 10^1$$
$$\cong 0.31416 \times 10^1$$
$$= fl(\pi)$$

Defn If p^* is an approximation to p, then the **absolute error** is $|p - p^*|$ and the **relative error** is $\dfrac{|p - p^*|}{|p|}$, provided $p \neq 0$.

Ex Let $p = 0.3000 \times 10^1$ and $p^* = 0.3100 \times 10^1$. Then

$$\text{abs error} = |0.3000 \times 10^1 - 0.3100 \times 10^1|$$
$$= 0.0100 \times 10^1$$
$$= 0.1$$

$$\text{rel error} = \frac{0.1}{0.3000 \times 10^1}$$

$$= 0.333\overline{3} \times 10^{-1}$$

Ex Let $p = 0.3000 \times 10^{-3}$ and $p^* = 0.3100 \times 10^{-3}$. Then

$$\text{abs error} = |0.3000 \times 10^{-3} - 0.3100 \times 10^{-3}|$$
$$= 0.01 \times 10^{-3}$$
$$= 0.1 \times 10^{-4}$$

$$\text{rel error} = \frac{0.1 \times 10^{-4}}{0.3000 \times 10^{-3}} = 0.3333 \times 10^{-1}$$

Ex Let $p = 0.3000 \times 10^{4}$ and let $p^* = 0.3100 \times 10^{4}$. Then

$$\text{abs error} = |0.3000 \times 10^{4} - 0.3100 \times 10^{4}|$$
$$= 0.01 \times 10^{4}$$
$$= 0.1 \times 10^{3}$$

$$\text{rel error} = \frac{0.1 \times 10^{3}}{0.3000 \times 10^{4}} = 0.3333 \times 10^{-1}$$

Thus abs error can be misleading, while rel error takes into account size of value p.

0.3012

3.012×10^{-1}

$< 5 \times 10^{-1}$

0.5012

$= 5.012 \times 10^{-1}$

$= 0.5012 \times 10^{0}$

$< 5 \times 10^{0}$

Defn The number p^* is said to approximate p to t significant digits (or figures) if t is the largest nonnegative integer for which

$$\text{rel error} = \frac{|p - p^*|}{|p|} < 5 \times 10^{-t}$$

Ex Suppose $t = 4$ (4 significant digits); that is, p^* agrees with p to four significant digits. Consider table below

p	0.1	0.5	1000	9990
$\max\lvert p - p^* \rvert$	0.00005	0.00025	0.5	4.9985

4.995

$$|p - p^*| < |p|[5 \times 10^{-4}]$$
$$= 0.1 \cdot [5 \times 10^{-4}]$$
$$= 0.5 \times 10^{-4}$$
$$= 5 \times 10^{-5}$$

$$p - (5 \times 10^{-5}) < p^* < p + (5 \times 10^{-5})$$

or $p^* \in (p - 5 \times 10^{-5}, p + 5 \times 10^{-5})$

$$|p - p^*| < |p| \cdot [5 \times 10^{-4}]$$
$$= 9990[5 \times 10^{-4}]$$
$$= 49.950 \times 10^{-4}$$
$$= 4.9950$$

$p^* \in (p - 4.9950, p + 4.9950)$

17

Ex Recall that when $p = 0.3000 \times 10^4$ and $p^* = 0.3100 \times 10^4$, we found

$$\text{rel error} = \frac{|p - p^*|}{|p|} = 0.333\overline{3} \times 10^{-1}$$

Thus $$\text{rel error} = 3.333\overline{3} \times 10^{-2}$$
$$< 5 \times 10^{-2}$$

Hence p^* approximates p to 2 significant digits.

Note that it is also true that
$$\text{rel error} < 5 \times 10^{-1}$$
but we would not say that p^* approx's p to 1 significant digit since 1 is not the largest nonnegative integer exponent on 10 for which this inequality holds (2 is). Similarly, 3 significant digits is incorrect since $\text{rel error} \not< 5 \times 10^{-3}$.

The relative error for the floating point representation $fl(y)$ of y is

$$\left|\frac{y - fl(y)}{y}\right|$$

If k decimal digits and chopping are used for the machine representation of $y = 0.d_1 d_2 \cdots d_k d_{k+1} \cdots \times 10^n$, then

$$\left|\frac{y - fl(y)}{y}\right| = \left|\frac{0.d_1 d_2 \cdots d_k d_{k+1} \cdots \times 10^n - 0.d_1 d_2 \cdots d_k \times 10^n}{0.d_1 d_2 \cdots d_k d_{k+1} \cdots \times 10^n}\right|$$

$$= \left|\frac{0.\overbrace{00\cdots0}^{k \text{ zeros}} d_{k+1} d_{k+2} \cdots \times 10^n}{0.d_1 d_2 \cdots d_k d_{k+1} \cdots \times 10^n}\right|$$

Moving k places to left is compensated by multiplying by 10^{-k}

$$= \left|\frac{0.d_{k+1} d_{k+2} \cdots \times 10^{n-k}}{0.d_1 d_2 \cdots d_k d_{k+1} \cdots \times 10^n}\right|$$

Cancel the 10^n

$$= \left|\frac{0.d_{k+1} d_{k+2} \cdots}{0.d_1 d_2 \cdots d_k d_{k+1} d_{k+2} \cdots}\right| \times 10^{-k}$$

$$\leq \frac{0.999\cdots}{0.100\cdots} \times 10^{-k} \leq \frac{1}{0.1} \times 10^{-k} = 10^{-k+1}$$

∎

Thus an upper bound for relative error of $fl(y)$, using k-digit chopping, is 10^{-k+1}:

$$\left| \frac{y - fl(y)}{y} \right| \leq 10^{-k+1}$$

For k-digit rounding, we have

$$fl(y) = Chop_k(0.d_1d_2 \cdots d_k d_{k+1} \cdots \times 10^n + 5 \times 10^{n-(k+1)})$$

$$= Chop_k(0.d_1d_2 \cdots d_k d_{k+1} \cdots \times 10^n + 0.\underbrace{00\ldots00}_{k \text{ zeros}}5 \times 10^n)$$

$$= Chop_k(0.d_1d_2 \cdots d_k (d_{k+1}+5) d_{k+2} \cdots \times 10^n)$$

(d_{k+2})

If $d_{k+1} < 5$, then $fl(y) = 0.d_1d_2 \cdots d_k \times 10^n$

If $d_{k+1} \geq 5$, then

$$fl(y) = 0.d_1d_2 \cdots d_k * 10^n + 0.\overbrace{00\cdots01}^{k-1 \text{ zeros}} \times 10^n$$

$$= 0.d_1d_2 \cdots d_k \times 10^n + 10^{n-k}$$

(since $0.\underbrace{00\cdots01}_{k \text{ decimal places}} = 10^{-k}$)

Suppose first that $d_{k+1} < 5$. Then

$$\left|\frac{y - fl(y)}{y}\right| = \left|\frac{[0.d_1d_2\cdots d_kd_{k+1}\cdots \times 10^n] - [0.d_1d_2\cdots d_k \times 10^n]}{0.d_1d_2\cdots d_kd_{k+1}\cdots \times 10^n}\right|$$

$$= \left|\frac{0.d_{k+1}d_{k+2}\cdots \times 10^{n-k}}{0.d_1d_2\cdots d_kd_{k+1} \times 10^n}\right|$$

$$\leq \frac{0.5 \times 10^{-k}}{0.1} = \underline{0.5 \times 10^{-k+1}}$$

Now suppose $d_{k+1} \geq 5$. Then

$$\left|\frac{y - fl(y)}{y}\right| = \left|\frac{[0.d_1d_2\cdots d_kd_{k+1}\cdots \times 10^n] - [0.d_1d_2\cdots d_k \times 10^n + 10^{n-k}]}{0.d_1d_2\cdots d_k d_{k+1}\cdots \times 10^n}\right|$$

$$= \left|\frac{(0.d_{k+1}d_{k+2}\cdots) \times 10^{n-k} - 10^{n-k}}{(0.d_1d_2\cdots d_kd_{k+1}\cdots) \times 10^n}\right|$$

$$= \left|\frac{(0.d_{k+1}d_{k+2}\cdots) - 1}{0.d_1d_2\cdots d_kd_{k+1}\cdots}\right| \times 10^{-k}$$

$$\leq \left|\frac{0.5 - 1}{0.1}\right| \times 10^{-k} = \underline{0.5 \times 10^{-k}}$$

Thus an upper bound for the relative error when using k-digit rounding is $0.5 \times 10^{-k+1}$:

$$\left| \frac{fl(y) - y}{y} \right| \leq 0.5 \times 10^{-k+1}$$

Additional errors are introduced when we perform computations with $fl(y)$. That is, suppose x & y are real numbers, with computer representations given by $fl(x)$ and $fl(y)$.

Also, denote exact arithmetic operations by $+, -, \times$ and $\div$. Denote finite precision (computer) arithmetic operations by $\oplus, \ominus, \otimes$ and $\oslash$.

Finite-digit arithmetic can be assumed to be given by

$$x \oplus y = fl(fl(x) + fl(y))$$

$$x \ominus y = fl(fl(x) - fl(y))$$

$$x \otimes y = fl(fl(x) \times fl(y))$$

$$x \oslash y = fl(fl(x) \div fl(y))$$

That is, first perform exact arithmetic on floating point numbers, and then convert exact result to its finite digit floating point representation (i.e., round or chop result).

Ex Let $x = \frac{5}{7}$, $y = \frac{1}{3}$, and suppose we use 5-digit chopping.

Then $fl(x) = 0.71428 \times 10^0$

and $fl(y) = 0.33333 \times 10^0$

It follows that

$$x \oplus y = fl(fl(x) + fl(y))$$
$$= fl(0.71428 \times 10^0 + 0.33333 \times 10^0)$$
$$= fl(1.04761 \times 10^0)$$
$$= fl(0.104761 \times 10^1)$$
$$= 0.10476 \times 10^1$$

The actual value is

$$x + y = \frac{5}{7} + \frac{1}{3} = \frac{15}{21} + \frac{7}{21} = \frac{22}{21}$$
$$= 1.0476190\ldots$$

(Note that $fl(x+y) = 1.0476 \times 10^0 = 0.10476 \times 10^1$)

Abs Error $= 0.19 \times 10^{-4}$

Rel Error $= 0.182 \times 10^{-4} = \frac{|(x+y) - (x \oplus y)|}{|x+y|}$

See Table 1.2 for results of similar calculations $x \ominus y$, $x \oslash y$, $x \oplus y$. The maximum relative error is 0.267×10^{-4}, and hence the arithmetic process produces satisfactory five-digit results.

Suppose next that we compute these arithmetic operations for

$$u = 0.714251, \quad V = 98765.9, \quad w = 0.111111 \times 10^{-4}$$

Then

$$fl(u) = 0.71425 \times 10^{0}$$
$$fl(v) = 0.98765 \times 10^{5}$$
$$fl(w) = 0.11111 \times 10^{-4}$$

These numbers were chosen to illustrate some problems that can arise with finite-digit arithmetic. See Table 1.3 for a summary of results for various arithmetic computations.

Table 1.2

Operation	Result	Actual value	Absolute error	Relative error
$x \oplus y$	0.10476×10^{1}	22/21	0.190×10^{-4}	0.182×10^{-4}
$x \ominus y$	0.38095×10^{0}	8/21	0.238×10^{-5}	0.625×10^{-5}
$x \otimes y$	0.23809×10^{0}	5/21	0.524×10^{-5}	0.220×10^{-4}
$y \oplus x$	0.21428×10^{1}	15/7	0.571×10^{-4}	0.267×10^{-4}

Table 1.3

Operation	Result	Actual value	Absolute error	Relative error
$x \ominus u$	0.30000×10^{-4}	0.34714×10^{-4}	0.471×10^{-5}	0.136
$(x \ominus u) \oplus w$	0.29629×10^{1}	0.34285×10^{1}	0.465	0.136
$(x \ominus u) \otimes v$	0.29629×10^{1}	0.34285×10^{1}	0.465	0.136
$u \oplus v$	0.98765×10^{5}	0.98766×10^{5}	0.161×10^{1}	0.163×10^{-4}

$x = 5/7$, $fl(x) = 0.71428 \times 10^{0}$

$y = 1/3$, $fl(y) = 0.33333 \times 10^{0}$

$u = 0.714251$, $fl(u) = 0.71425 \times 10^{0}$

$v = 98765.9$, $fl(v) = 0.98765 \times 10^{5}$

$w = 0.1111111 \times 10^{-4}$, $fl(w) = 0.11111 \times 10^{-4}$

Note that $x \ominus u$ results in a small absolute error but a large relative error.

Dividing $x \ominus u$ by the small number w results in an increase in the absolute error while relative error stays the same.

Similarly when multiplying $x \ominus u$ by large v.

The addition of large & small numbers u & v produces large absolute error but not large relative error.

Cancellation Error Examples

Example 1: Consider a 4 Digit Calculator

Calculator:

$$\frac{(1+0.2345)-(1.23+0.0044)}{0.0001}=\frac{1.235-1.234}{0.0001}=\frac{0.001}{0.0001}=10$$

True Value:

$$\frac{(1+0.2345)-(1.23+0.0044)}{0.0001}=\frac{1.2345-1.2344}{0.0001}=\frac{0.0001}{0.0001}=1$$

What went wrong: First, the calculator has an inaccurate digit (the "5" in 1.235) due to round-off error. As a consequence, the "0.001" in the numerator is incorrect. Even worse, the "0.001" in the numerator has no accurate significant digits (the zeros in front of the "1" are place holders and don't count as significant digits). This loss of accurate digits is known as *cancellation error*.

Cancellation error occurs when subtracting relatively big numbers that differ only by a tiny amount, and always results in loss of many accurate digits. This cancellation error is made larger when dividing by the small number 0.0001. Two more examples are given below, for calculators that are often used on homework and exams.

Example 2: The TI-83/86 has 14 digit accuracy

Calculator:

$$(1{,}000{,}000 + 0.12341234) - (1{,}000{,}000 + 0.1234123) = 0$$

No accurate significant digits in result, and up to eight accurate significant digits given. In fact, there are 14 accurate significant digits in results of each sum on the left side of the equation, before subtracting.

True Value:

$$(1{,}000{,}000 + 0.12341234) - (1{,}000{,}000 + 0.1234123) = 0.00000004$$

One accurate significant digit in result, and up to eight accurate significant digits given.

Example 3: The TI-83/86 has 14 digit accuracy

Calculator:

$$\frac{(1{,}000{,}000 + 0.12312345) - (1{,}000{,}000 + 0.12312344)}{0.00000001} = 10$$

True Value:

$$\frac{(1{,}000{,}000 + 0.12312345) - (1{,}000{,}000 + 0.12312344)}{0.00000001} = \frac{0.00000001}{0.00000001} = 1$$

Cancellation Error

When two nearly equal numbers are subtracted, many significant digits can be lost.

Suppose x and y are nearly equal with $x > y$ and k-digit representations

$$fl(x) = 0.d_1 d_2 \cdots d_p \alpha_{p+1} \alpha_{p+2} \cdots \alpha_k \times 10^n$$
$$fl(y) = 0.d_1 d_2 \cdots d_p \beta_{p+1} \beta_{p+2} \cdots \beta_k \times 10^n$$

Then

$$x \ominus y = fl(fl(x) - fl(y))$$
$$= [(0.\alpha_{p+1} \alpha_{p+2} \cdots \alpha_k) - (0.\beta_{p+1} \beta_{p+2} \cdots \beta_k)] \times 10^{n-p}$$

(β_{p+1} and 10^{n-p} annotated)

Thus $x \ominus y$ has at most $k-p$ digits of significance.

On the previous slide, we found that $x \ominus y$ will have at most k-p digits of significance.

Since the computer uses k-digits to represent numbers, $fl(x-y) = x \ominus y$ will be assigned k digits, with the last p digits being either zero or randomly assigned.

Any further calculations involving $x-y$ retain the problem of only having k-p digits of significance.

If a finite digit representation or calculation introduces an error, further enlargement of the error occurs when dividing by a number with small magnitude, or when multiplying by a large number.

To help quantify this, suppose z has $fl(z) = z + \delta$, where the error δ is introduced by representation or previous calculation.

If we divide z by $\varepsilon = 10^{-n}$, $n > 0$, then

$$\frac{z}{\varepsilon} \approx fl\left(\frac{fl(z)}{fl(\varepsilon)}\right) = fl\left(\frac{z+\delta}{10^{-n}}\right)$$

$$= (z+\delta) \times 10^{n}$$

The absolute error is then

$$\left|\frac{z}{\varepsilon} - (z+\delta)\times 10^{n}\right| = \left|z \times 10^{n} - (z+\delta)\times 10^{n}\right|$$

$$= |\delta| \times 10^{n},$$

which is the original error multiplied by 10^n!

Note:

$$\text{rel error} = \frac{|\delta| \times 10^{n}}{|z/\varepsilon|} = \frac{|\delta| \times 10^{n}}{|z| \times 10^{n}} = \frac{|\delta|}{|z|}$$

$$\left(\text{and } \left|\frac{z-(z+\delta)}{z}\right| = \frac{|\delta|}{|z|}\right)$$

Ex Let $p = 0.54617$ and $q = 0.54601$. The exact value of $r = p - q$ is $r = 0.00016$. Using 4 digit rounding we have

$$p^* = 0.5462 \,, \quad q^* = 0.5460$$

and $r^* = p^* - q^* = 0.0002$.

Now

$$\text{relative error} = \frac{|r - r^*|}{|r|} =$$

$$= \left| \frac{0.00016 - 0.0002}{0.00016} \right|$$

$$= 0.25$$

Since $0.25 < 5 \times 10^{-1}$, it follows that r^* has only one significant digit, even though p^* and q^* were accurate to ~~three~~ four and ~~four~~ five digits, respectively.

$\left(\left| \frac{p - p^*}{p} \right| \approx 5.5 \times 10^{-5} < 5 \times 10^{-4} \,; \quad \left| \frac{q - q^*}{q} \right| \approx 1.8 \times 10^{-5} < 5 \times 10^{-5} \right)$

$5.5 \times 10^{-5} < 0.55 \times 10^{-4}$

Ex The roots of $ax^2+bx+c=0$, $a\neq 0$, are given by

$$x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}, \quad x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$$

Suppose we want to solve

$$x^2+62.10x+1=0,$$

whose roots are approximately

$$x_1 = -0.01610723, \quad x_2 = -62.08390$$

Using the quadratic formula with 4-digit rounding, we have

$$\sqrt{b^2-4ac} = \sqrt{(62.10)^2-(4.000)(1.000)(1.000)}$$

$$\cong \sqrt{3856-4.000} = \sqrt{3852} \cong 62.06$$

Since $b^2 >> 4ac$, cancellation error will occur in computing x_1:

$$fl(x_1) = \frac{-62.10+62.06}{2.000} = \frac{-0.0400}{2.000} = -0.0200$$

$$\text{relerror} = \frac{|-0.01610723+0.02000|}{0.01610723} \cong 0.2417$$

← one significant digit of accuracy

Calculation of x_2 does not involve cancellation, since we are adding two nearly equal numbers.

$$fl(x_2) = \frac{-62.10 - 62.06}{2.000} = \frac{-124.2}{2} = -62.10$$

$$\text{rel error} = \left| \frac{-62.08390 + 62.10}{-62.08390} \right| \cong 0.00026$$

four significant digits of accuracy $\longleftrightarrow$ $= 2.6 \times 10^{-4}$

To obtain a more accurate 4 digit approximation for x_1, rationalize numerator:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \left(\frac{-b - \sqrt{b^2 - 4ac}}{-b - \sqrt{b^2 - 4ac}} \right) = \frac{b^2 - (b^2 - 4ac)}{2a(-b - \sqrt{b^2 - 4ac})}$$

Thus

$$x_1 = \frac{-2c}{b + \sqrt{b^2 - 4ac}}$$

using this formula,

$$fl(x_1) = \frac{-2.000}{62.10 + 62.06} = \frac{-2}{124.2} = -0.0161,$$

rel error $\cong 6.2 \times 10^{-4}$ (3 significant digits of accuracy)

The rationalization technique can also be applied to x_2:

$$x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}\left(\frac{-b+\sqrt{b^2-4ac}}{-b+\sqrt{b^2-4ac}}\right)$$

$$= \frac{b^2-(b^2-4ac)}{2a(-b+\sqrt{b^2-4ac})} = \frac{2c}{-b+\sqrt{b^2-4ac}}$$

If $b<0$, then this is the form to use. Otherwise, not only will cancellation occur in the denominator (when $b^2 \gg 4ac$) but we will subsequently be dividing by a small number, hence enlarging the effect of the inaccuracy. Here,

$$fl(x_2) = \frac{2c}{-b+\sqrt{b^2-4ac}} = \frac{2.000}{-62.10+62.06} = \frac{-2.000}{0.0400} = -50.0$$

rel error $\cong 1.9 \times 10^{-1}$ (One significant digit of accuracy)

In general, accuracy loss due to roundoff error can be reduced by rearranging calculations.

Ex Evaluate $f(x) = x^3 - 6.1x^2 + 3.2x + 1.5$ at $x = 4.71$ using three-digit arithmetic.

Intermediate results:

	x	x^2	x^3	$6.1x^2$	$3.2x$
Exact	4.71	22.1841	104.487111	135.32301	15.072
3 digit chopping	4.71	22.1	104.	134.	15.0
3digit rounding	4.71	22.2	105.	135.	15.1

Calculating $f(4.71)$:

Exact: $f(4.71) = -14.263899$

Chopping: $f(4.71) \cong -13.5$

rel error $\cong 0.05 = 0.5 \times 10^{-1}$ (1 sig. digit)

Rounding: $f(4.71) \cong -13.4$

rel error $\cong 0.06 = 0.6 \times 10^{-1}$ (1 sig. digit)

Alternatively, write f in nested form:

$$f(x) = x^3 - 6.1x^2 + 3.2x + 1.5 = [(x - 6.1)x + 3.2]x + 1.5$$

Then

3-Chopping: $f(4.71) \cong -14.2$, rel error $\cong 0.0045$

3-Rounding: $f(4.71) \cong -14.3$, rel error $\cong 0.0025$

} 3 signif. digits accuracy

Recall from previous slide:

$$f(x) = x^3 - 6.1x^2 + 3.2x + 1.5 = [(x - 6.1)x + 3.2]x + 1.5$$

Polynomials should always be expressed in nested form before performing evaluation, because this form minimizes the number of arithmetic calculations.

The decreased error in the example above is due to reduction in computations from 4 multiplications and three additions to 2 multiplications and 3 additions.

Thus one way to reduce roundoff error is to reduce the number of error producing computations.